Graph Transformation Model of a Triangulated Network of Mobile Units

Stefan Gruner

Dept. of Comp. Sc. | Univ. of Pretoria | 0002 Pretoria | South-Africa | sg@cs.up.ac.za

Abstract: A triangulated network of mobile units is modelled by means of a graph transformation system in which graph nodes are labelled with geometric coordinates and edges are labelled with distances. Nodes represent mobile units and edges represent wireless radio communication links between them. Under concurrency the model can describe interesting practical scenarios, for example swarms of taxis in an urban environment. The contribution features the enhancement of a graph transformation system by trigonometric calculations. By the way it is also shown that the classical "negative edge condition" has only limited applicability if a strict locality principle is assumed, and –vice versa– that there are reasonable modeling cases in which this locality principle itself fails to suffice.

Keywords: Attributed Graph Transformation, Mobile Network, Concurrency, Locality.

1 Scenario

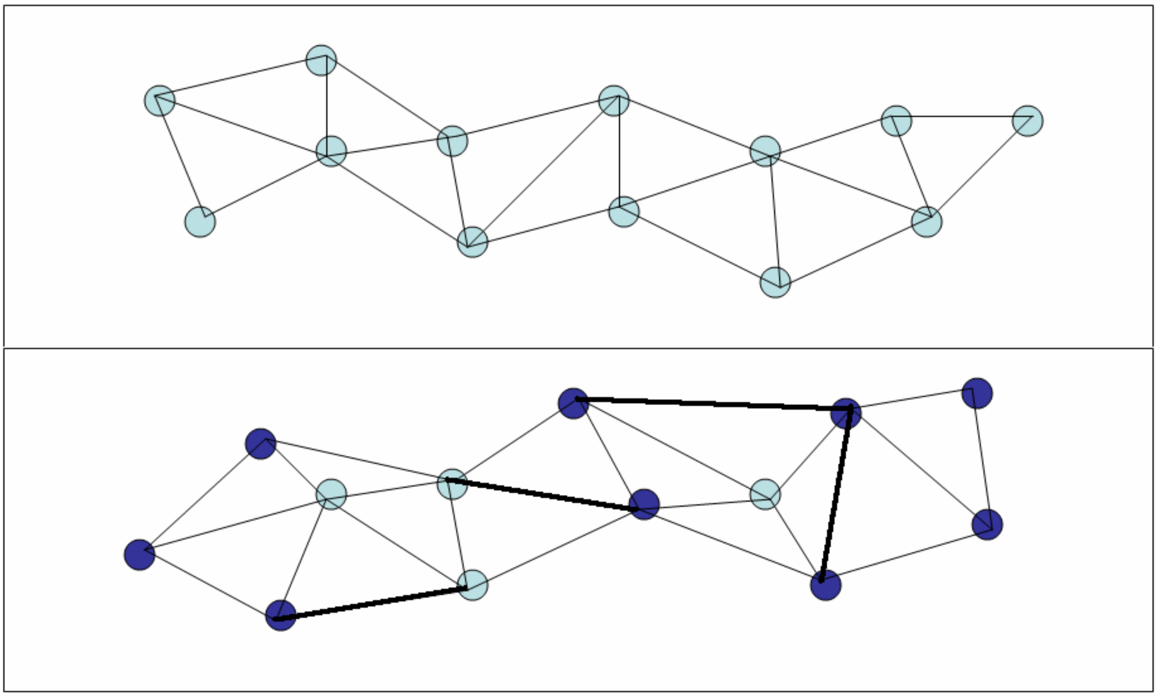

Figure 1: *Triangulated network before* (top) *and after* (bottom) *the movement of some nodes. Dark shaded nodes have changed their positions, and some communication links have also been reconfigured in their topological positions, which is depicted by thicker lines.*

Imagine an inner-city scenario in which a swarm of independent yet cooperating taxis keep each other informed about sources of customers to be picked up, traffic jams in the streets, etc. Every taxi is equipped with a simplistic communication device to keep in contact with a small number of other units in a not-too-far distance. To avoid confusion between the units, two *locality constraints* are imposed on every unit:

- The number *N* of communication partners to each unit is limited.
- The communication distance *D* between each unit is limited, too.

The network is self-organizing like a swarm of fish, thus not controlled by any central agency. Furthermore, the network structure shall be *triangulated* (as further defined below), such that:

- It has a well-manageable regular internal structure, as depicted in Figure 1, whereby:
- Configurations can be simply (re)-calculated with the usual formulae of trigonometry.

In the following sections of this paper a simple yet effective graph transformation model to such a self-organizing mobile network is developed. The underlying graph transformation techniques themselves being rather "classical" (except of a new interpretation of the "negative edge" condition) the value of this study is to be found in its general and uniform representation of a practically relevant scenario – enabling simulative experiments to study the behaviour of such systems, locally and globally, under various settings of its key parameters.

2 Technical Preliminaries

The *graph transformation paradigm* used in this paper combines the PROGRES system's syntax [SWZ99] and denotational semantics [Sch96] of program-embedded attributed graph transformation with the VISIDIA system's *message-passing* operational semantics [BGM01] [LMS95] as follows:

- Edges in the left-hand-side of a graph transformation rule represent communication link between two units, whereby the liveness of such links must be acknowledged by means of message-passing between the connected units.
- The *non*-existence of an edge between two units *u* and *u'* (negative edge condition) can thus only be recognized *indirectly* via a third unit *u"* to which both *u* and *u'* are linked. Any rule with a left-hand-side combining only *u* and *u'* with a "negative edge" (which would be perfectly legal in PROGRES with its "omnivident" global viewpoint) is thus *meaningless* under this strictly local operational perspective.
- An edge decorated with an explicit edge-attribute *e* indicates that some information *e* is (or can be) shared between the two adjacent nodes by means of message-passing as described above. Edge-attributes are only "syntactic sugar" for the sake of legibility of the model specification – in the terminology of PROGRES: "derived" (not "intrinsic").
- The operational semantics (implementation) of an edge between an ordinary node *u* and a multi-node *u** (which represents a finite set of nodes in the neighbourhood of *u*) requires *star-synchronisation* under *mutual exclusion,* as explained in [BGM01].
- Where mutual exclusion is not locally required, arbitrarily many transformation rules may be applied across the network graph at any time, in any paradigm of concurrency; see for example [LMS99].
- *Isomorphisms* are generally used to map the transformation rules' left-hand-sides into the model graph for application – with special treatment of the multi-nodes [SWZ99].

The *triangulation property* of the network graphs in this paper is defined as follows: A *ring* of size r is a graph with r nodes $\{V_0,\ldots V_{r-1}\}$ and r edges $\{E_0,\ldots E_{r-1}\}$ such that for all $0 \leq i < r$: $E_i = (V_i, V_{i+1 \bmod r})$ and no further edges exist between those nodes than just these ones. Then a *planar* graph is called *triangulated* if all its ring-shaped sub-graphs are of size $r = 3$. Thereby it is not required that all possibilities of triangle-building are fully exhausted: As it can be seen in Figure 1, there might be "incomplete" triangle subgraphs $\{V_x, V_y, V_z\}$ with edges (V_x, V_y) and (V_y, V_z) but no ring-closing edge (V_z, V_x).

3 Model Specification

In the following, the terms "attribute" (from the PROGRES terminology) and "label" (from the VISIDIA terminology) are used synonymously. In the model developed in this paper for the mobile units scenario, nodes and edges of the network graph are labelled as follows:

- An edge can carry a variable label *D*, representing a geometric distance between two nodes (in whatever vector-space, naïvely two-dimensional Euclidean). Also remember what has generally been said about edge labels in the previous section.
- Nodes carry several labels of the following kind:
    - *C* is a vector-type label representing the coordinates (position) of the node in the chosen vector-space (environment).
    - *A* or *P* are auxiliary labels (modelling artefacts without a corresponding property in the modelled world), used for the treatment of concurrency and mutual exclusion as further explained below.
    - A label *OFF* is used to model defective nodes (units) which cannot communicate. In the absence of this label the unit is assumed to be operative; nodes with edges cannot be *OFF*.
    - A label *NEW* can be used to signify previously non-existing units which have just arrived in the operational terrain covered by the network, or also to signify the recovery of a previously defective node. New nodes cannot be connected immediately – in other words: nodes with edges cannot be of type *NEW*.

Consequently the graph transformation rules of the model are labelled, too. However, there are also *generic* rules in which some of the labels (node labels or edge labels) are omitted. This means that such a rule can be applied without taking the instance-value of the omitted label into account. For example, if a graph transformation rule depicts an edge without edge label *D* then this edge could be mapped to any edge in the model graph. Thus, the model is based on a simple hierarchic type system, with *ANY* being the only super-type to all the other concrete types and no intermediate types in between. Omission of a type is regarded as equivalent to the explicit labelling of the according entity with the *ANY* symbol – see [Sch97] for comparison.

3.1 Locally Mutual Exclusion of Activities

The first rule of the model checks if a node, which intends to change its geographic position, is free to do so. If this is the case, it takes a token which prevents any immediate neighbour from becoming active as well. As usual it is assumed that the choice of an anchor-place for rule application is made non-deterministically (at random). In Figure 2 this rule is depicted. It tells us that a node can pick an activity token (for mutual exclusion in the local neighbourhood) if this node itself as well as all its adjacent nodes are currently passive. As mentioned above, the model rules shall be *denotationally* interpreted like in PROGRES, whereby an additional star symbol (*) is used here as “lexical sugar” to emphasize the mapping of a multi-node to the entire neighbourhood (star) of a central node (here: node 1). Moreover it must be kept in mind how two overlapping rule-applications in the same local neighbourhood are prevented by the *operational* semantics of the VISIDIA system [BGM01][LMS99], due to which it can never happen that two adjacent units find themselves both in an active state *A* at the very same time. This means that simultaneous activities of two immediately adjacent units in the scenario are modelled via pseudo-simultaneous interleaving of mutually exclusive actions in rapid pace.

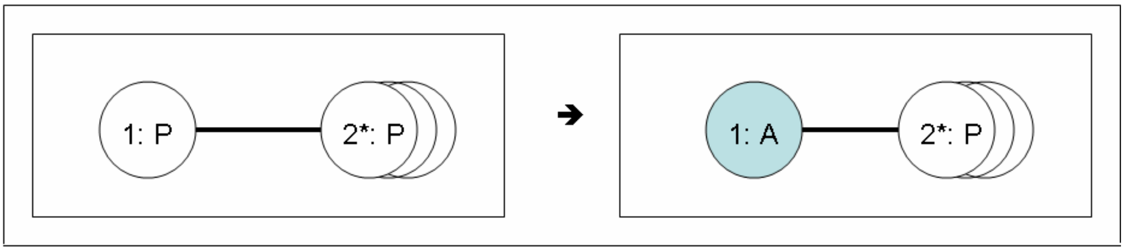

Figure 2: *Rule which explicitly models mutual exclusion of activities in a neighbourhood. The passive centre node of a star may pick an activity-token if all its neighbours are passive too. An operational semantics like the one implemented in the VISIDIA system guarantees that this rule itself is not multiply applied in the same local neighbourhood at the same time, which prevents two adjacent nodes from taking the activity token A simultaneously.*

Thus, the issue of local mutual exclusion is treated in this paper at two different levels: *explicitly* in the model specification by means of the *A* and *P* labels, as well as *implicitly* by VISIDIA operational rule-application semantics (message exchange along the communication channels). In non-overlapping (remote) regions of the model graph, however, a rule may well be multiply applied, at the same time, in genuine non-interleaving concurrency. Technically speaking this explicit modelling of local mutual exclusion would not be necessary; the implicit star-synchronisation of a VISIDIA kind of operational semantics would suffice. Nevertheless the explicit exclusion model with labels *A* and *P* was chosen for the sake of conceptual clarity.

3.2 Movement of Units

After a node has acquired activity-status (see above) it can change its location according to the mobile unit scenario. This is modelled with node labels representing coordinates in the chosen vector space. “Motion” is thus nothing but node relabelling, interpreted in terms of the chosen domain. When a node has been “moved” (by relabelling), two further actions must take place:

- The adjacent edges, labelled with the distances between the active node and its passive neighbour nodes, must also get relabelled to correctly represent the new geographic configuration.
- Then, the active node must release its exclusion label *A* and return to a passive state *P*.

As shown below, all these actions can be expressed in one single graph transformation rule. For the recalculation of positions and distances, the graph grammar system is augmented with a simple trigonometric calculus which must be executed while the graph grammar rules are applied. The authors of PROGRES system have demonstrated how this can be done [SWZ99].

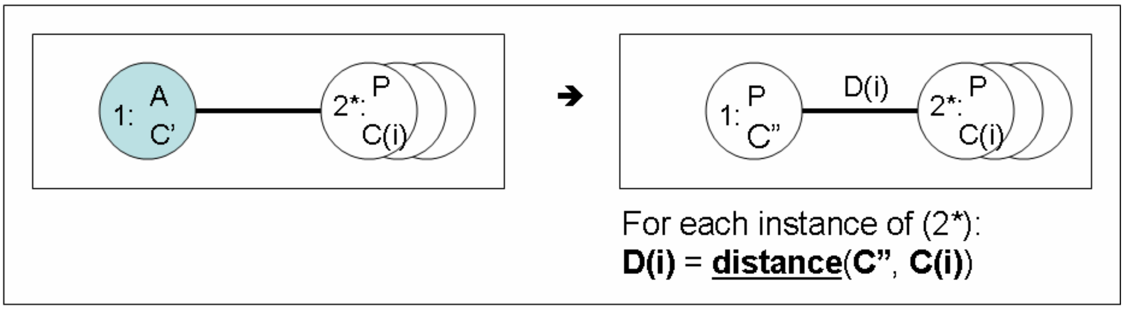

Figure 3: *Movement rule. The distance is calculated by an imported function, according to the PROGRES system. Thereafter the active node gives up its mutual exclusion token and returns into a passive state.*

Note that the movement of a node can temporarily destroy the desired triangulation property of the underlying network. For reasons of model simplicity (which means: a small number of small rules) it has been decided to temporarily concede the violation of this global topologic network property and to fix any violation with an equally simple set of repair-rules, rather than trying to design a complicated graph transformation system with large (non-local) rules for the sake of avoiding the violation of the triangulation property in the first place. (The repair-rules will be shown in the subsequent section.) Also note that the usage of large (non-local) rules would undermine the concept of a self-organizing network which is not controlled by a central agency. Small (local) rules, on the other hand, can be easily applied by the network nodes themselves as described in the literature to the concurrent graph relabeling paradigm [LMS99].

3.3 Communication Breakdown

In the mobile unit scenario it should be realistically assumed that communication can break down from time to time (which can also result in a temporary violation of the triangulation property and will also be treated by repair-rules as described in the subsequent section). In this paper any instance of a communication breakdown has one of the following three reasons; (see Future Work section below for further considerations):

- Communication breakdown due to a unit-internal technical defect;
- Communication breakdown due to too far distance between the communicating units;
- Communication breakdown because a unit cannot cope with too large numbers of communication partners any more; (the connectivity degrees of network nodes are assumed to be limited).

For all these cases, model rules are provided in the following. The activity status (*A* or *P*) of the nodes involved is irrelevant here, because the activity status is only a model artefact by means of which simultaneous movements of adjacent nodes are simulated (through small-step interleaving under mutual exclusion). Where motion is not involved at all, explicit distinction between *A*-nodes and *P*-nodes is obsolete as well.

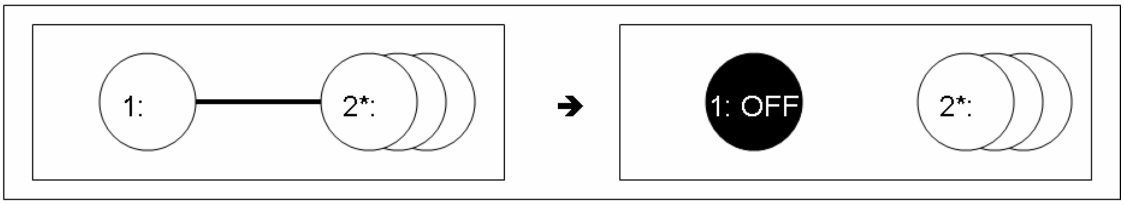

Figure 4: *Transformation rule describing a communication breakdown due to internal defect.*

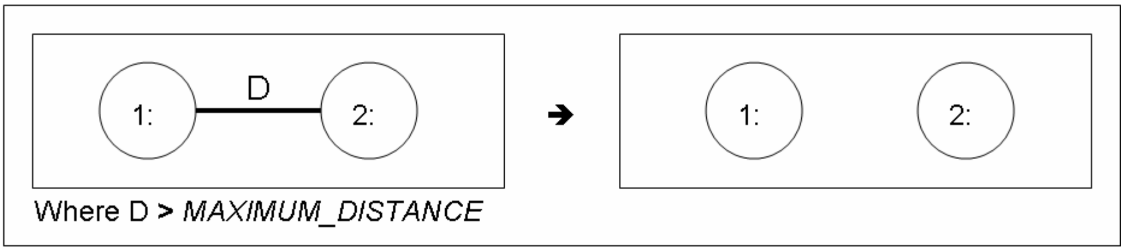

Figure 5: *Rule describing a communication breakdown due to long distance (weak signal).*

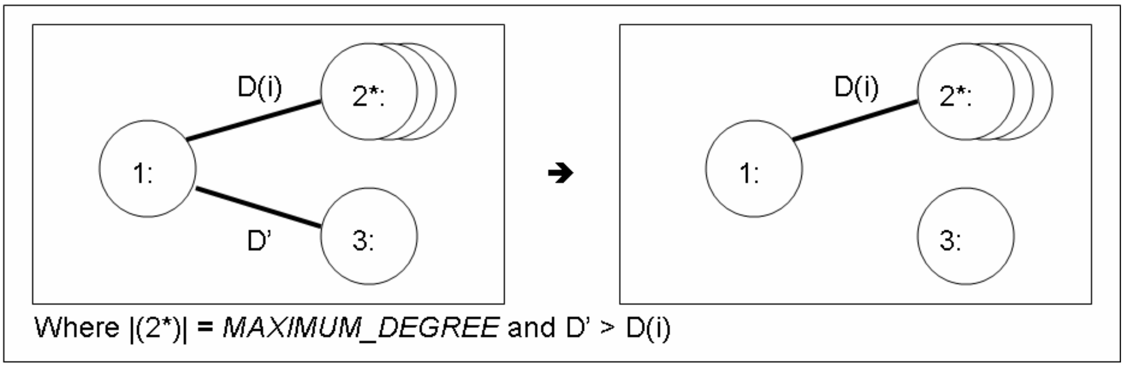

Figure 6: *Too many communication partners: the most remote one is abandoned. Classical graph transformation systems, without counting features, cannot process such kind of rules.*

As far as Figure 6 is concerned one might ask the question how a unit (here: 1) can have more than the maximal number of communication partners in the first place. The answer is found in the dynamic and self-optimizing nature of the system to be modelled: Temporary violations of global "invariant" properties (here: the maximal degree of connectivity) are admitted for a short period of time, such that (for example) a better communication link can be established before a worse one is given up. (This issue will become clearer in the subsequent paragraphs where the reparation of violated triangulations and the establishment of new communication links are modeled.) Also note that the rule depicted in Figure 6 requires a graph transformation paradigm which allows for *counting* the magnitude |N| of the neighbourhood of a centre node, which means that *neither* the classical PROGRES system [SWZ99] *nor* the classical VISIDIA system [BGM01] can be used to implement this crucial rule.

In this sub-section in only remains to be said that a defective (thus: isolated) node must be able to get back into operational mode, as a precondition to the re-establishment of its participation in the communicative network. This is modelled by the rather trivial rule depicted in Figure 7.

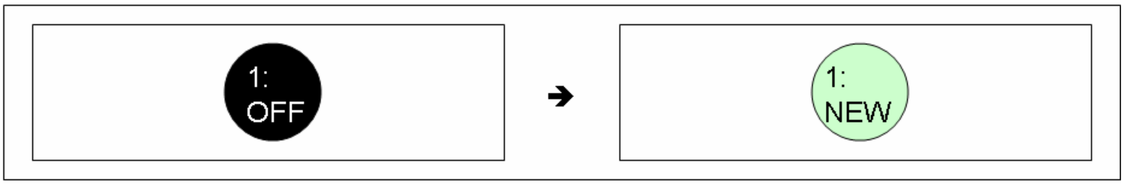

Figure 7: *A defective unit gets repaired. (It is as if a new unit would appear in the landscape.)*

3.4 Reparation of Locally-Temporarily Violated Network Invariants

The scenario model is generally assumed to possess a number of global invariants, such as the triangulated structure of the network graph, the maximal communication distance between two units, or the maximal degree of connectivity (number of communication partners) throughout the network. On the other hand it has already been mentioned that the dynamic character of the scenario will locally lead to violations of those invariants for short periods of time,[‡] until some repair-mechanisms restore the desired homogeneity of the model graph. These mechanisms are modelled explicitly (as part of the scenario specification) in the following paragraphs.

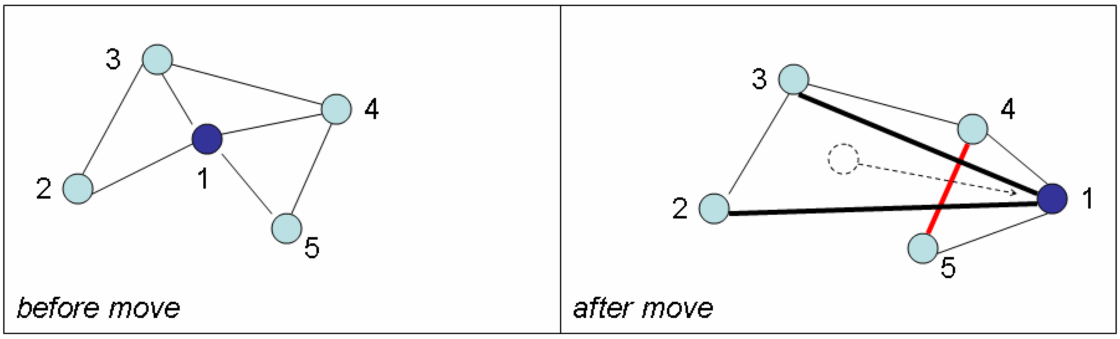

Figure 8: *Scenario in which the triangulation property is violated after a far-reaching move of unit 1 (see red line). The edges between units 1 and 2, respectively units 1 and 3, are now crossing the edge between units 4 and 5, such that this sub-graph is not planar any more.*

Consider the following situation depicted in Figure 8, in which some unit makes a move over some longer distance (by means of the motion rule of Figure 3) which leads to a local violation of the desired triangulation property. The strategy dealing with such situations consists of two subsequent steps:

- One of two edges which are "crossing" each other –which can only be described in terms of the vector-space geometry functions with which the graph transformation system is augmented– will be deleted; for the purpose of network optimization this is typically the longer link (with a greater distance label *D*).
- If such a deletion leads to a subsequent destruction of the network's triangulation property elsewhere, new communication links must be established at those locations, as it is sketched in Figure 9.

[‡] An analogy in the physical nature can be found in the realm of quantum physics, whereby spontaneous appearances of short-lived virtual particles can temporarily violate the macro-law of energy preservation for a short period of time in a small area of space.

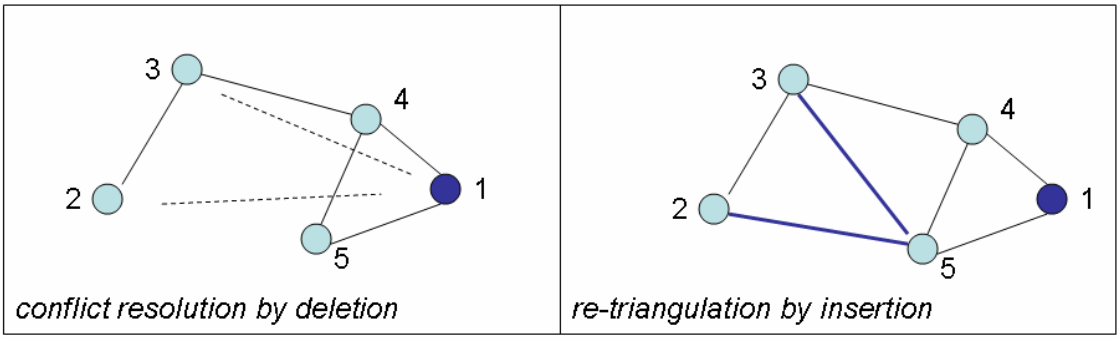
3
4
2
1
5
conflict resolution by deletion
3
4
2
1
5
re-triangulation by insertion

Figure 9: *Continuation of the scenario of Figure 8. In a first step* (left) *the conflicting links are deleted (see dotted lines). Thereafter* (right), *new links must be established elsewhere to re-establish the desired triangulation property (see thick blue lines).*

In the following, the graph transformation rules are shown by means of which scenarios like the one depicted in Figure 8 and Figure 9 can be effectively modelled. Note that this is the point of the specification at which the most complicated calculations in terms of the underlying geometry must be imported (in PROGRES style) into a graph transformation rule in order to confirm the existence of a cut-point between two finite lines in the underlying vector-space.

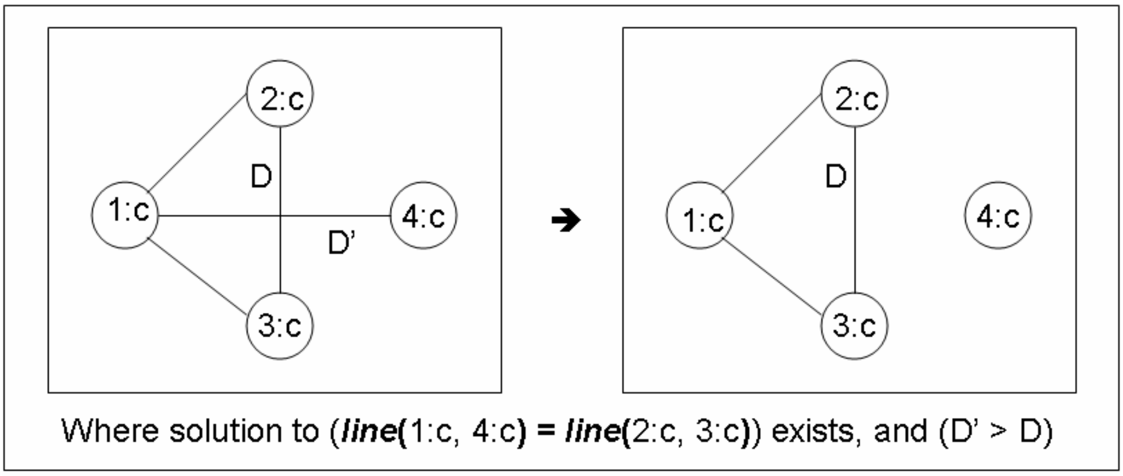
2:c
D
1:c
4:c
D'
3:c
➔
2:c
D
1:c
4:c
3:c
Where solution to (*line*(1:c, 4:c) = *line*(2:c, 3:c)) exists, and (D' > D)

Figure 10: *Graph rule describing the detection and deletion of a conflicting network link.*

In Figure 10 as well as in Figure 11 the "violated" triangle is formed by the nodes 1, 2 and 3. Note that the edges between nodes 1 and 2, respectively between 1 and 3, are *not* redundant for the detection of the cross-point, because these edges represent *communication* links: Node 1 is the central unit in this scenario, which detects the cross-point between line *D* and line *D'* by communication with its neighbours 2 and 3. A rule designed like the ones in Figure 10 and Figure 11, but *without* the edges between nodes 1 and 2, respectively 1 and 3, would imply the *global* (omnivident, communication-less) perspective of a classical graph transformation system such as PROGRES [SWZ99] – in contradiction to VISIDIA's operational semantics of locality and message-passing [BGM01] [LMS99], to which the model of this paper adheres. In this paradigm the left-hand-side of a rule with such a purpose can only be a connected graph.

Also note that node 4 of Figure 10 could *possibly* remain *isolated* after the according rule has been applied – ditto for the rules depicted in Figure 5 and Figure 6. In such a case the isolated node would assume a *NEW* state, like the one which is depicted in Figure 7. The according graph transformation rule is not depicted in this paper for reason of triviality: every unit knows intuitively (by itself) when it does not have any communication partners, thus when to relabel itself to *NEW*.

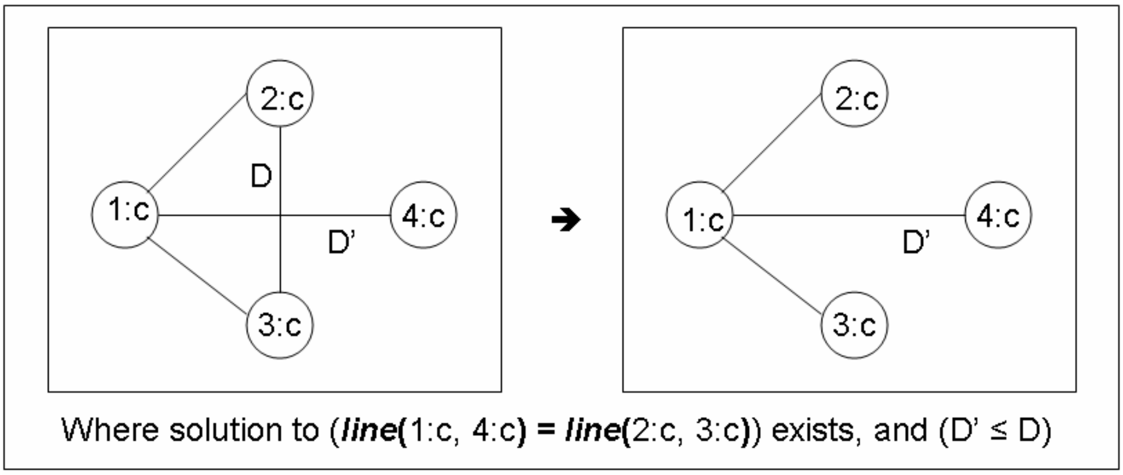
2:c
D
1:c
4:c
D'
3:c
➔
2:c
1:c
4:c
D'
3:c
Where solution to (*line*(1:c, 4:c) = *line*(2:c, 3:c)) exists, and (D' ≤ D)

Figure 11: *Ditto, this time with the other one of the two conflicting links being deleted.*

Further note that the rules of Figure 10 and Figure 11 together ensure *local optimization* by determining the *longer link to be deleted* and the shorter link to remain. It is obvious that two rules are needed for this optimization purpose: one for the case that $D > D'$, and one for the case that $D \leq D'$. These rules are designed as ordinary rules (not as a star-rules) such that only one edge in a conflict situation is deleted per rule application. This design option has been chosen in order to keep the amount of destruction within a region of the model graph as small as possible (thus also the amount of required restaurations) for conservative reasons. In case that more than one conflicting edge must be destroyed, these rules can be applied repeatedly until all conflicts are resolved.

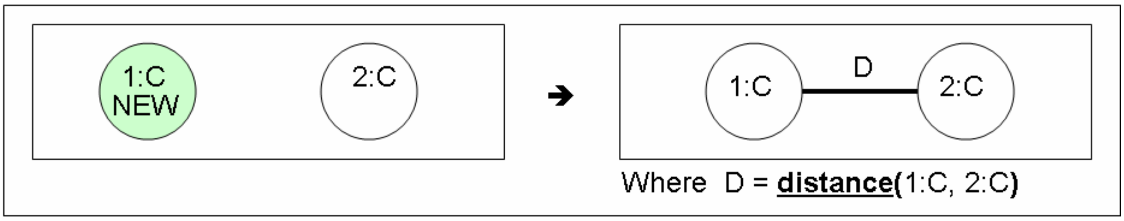

Figure 12: *Rule modeling a new unit receiving a radio signal and establishing a network link with its hitherto unknown newly found communication partner. No VISIDIA-semantics here!*

After an edge has been deleted as shown above, or after a defective unit has come back into operation mode, there are opportunities for the (re)-establishment of communication links for the sake of the already mentioned global triangulation property. In the case of a new node, which has per der definitionem no communication links (and therefore no knowledge about the existence of any other units) the new node cannot do anything but wait and listen until it receives a signal from another, hitherto unknown nearby unit. A communication link between these two units is then formally established, and the now connected node looses its *NEW* status, as depicted in Figure 12. Per default the value of *D* must be smaller than maximum, otherwise no radio signal could have been received at all. Should the newly established link violate any of the already mentioned global network invariants then the repair-rules (Figure 6, Figure 10, Figure 11) would be in place again to rectify the temporary topological flaw in an optimizing manner.

At this point it is important to note that the graph transformation rule of Figure 12 is the only rule of the scenario model which can *not* be explained in terms of VISIDIA's message-passing semantics. Here, and only here, it is necessary to assume the "omnivident" perspective of the PROGRES paradigm. The reason is that, from a strictly local perspective, nodes 1 and 2 of Figure 12 cannot know anything about each other unless a communication link exists – which is however not the case in the left-hand-side of that rule. From a local perspective, the event of receiving a signal from a hitherto unknown unit comes as an unpredictable surprise to the receiving unit. It is not possible to verify the existence of the left-hand-side situation of Figure 12 through message exchange, because any message exchange already implies the situation of the right-hand-side of Figure 12. When the software prototype to this scenario model is being implemented (see section Future Work), this problem must be solved *ad-hoc* by means of data structures which do *not* correspond to the "pure" VISIDIA theory; (possibly: representation of a unit's surrounding "landscape", storing information about the presence of radio signals).

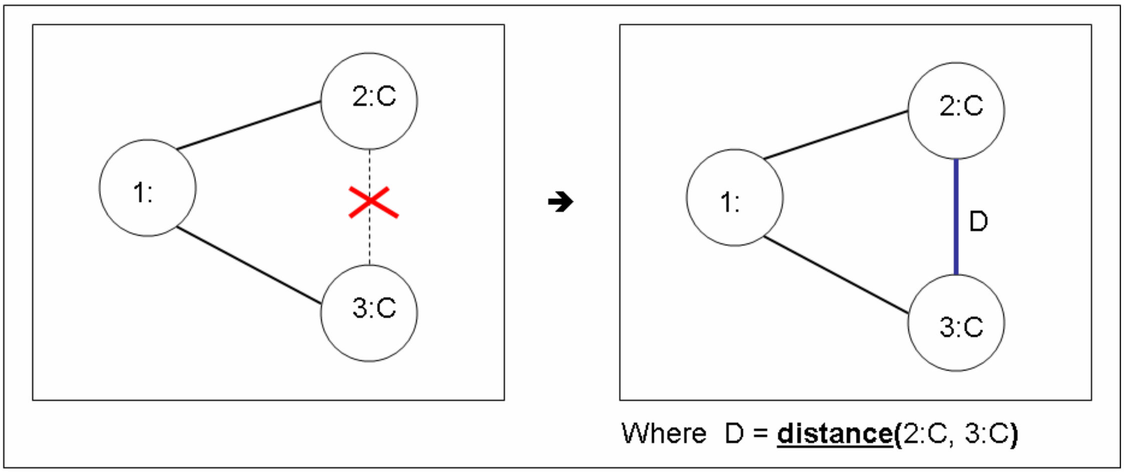

Figure 13: *Transformation rule describing the establishment of a new triangulation-link.*

Finally it must be explained how new "triangle" configurations in the model graph can come into existence, as it is shown in Figure 13. Per default, the distance *D* (between the newly connected nodes 2 and 3) cannot exceed the maximum. The dotted and crossed-out line in the left-hand side of the rule depicted in Figure 13, which models the establishment of a new triangulation, represents a *negative application condition* known from graph transformation systems such as PROGRES [SWZ99]. In the model of this paper (with its strict principle of locality in which the classical "omnividence" is no longer given), it must be explained how such an easily-drawn dotted line can be effectively implemented. This is done again by means of message passing (as explained in the Technical Preliminaries section above). Last but not least it should be mentioned that also the application of this rule (Figure 13) can lead to temporal violations of network invariants, but also in this case the already mentioned repair rules can be applied to fix the flaw.

3.5 Global Dynamics through Local Processes

All graph transformation rules in the model –except the one of Figure 12– are designed in such a way that they can be applied and executed by the network nodes themselves (without any global super-instance) with the techniques of the concurrent relabeling paradigm [BGM01]. Obviously no neighbourhood (star) radius is greater than 1 in these rules, which makes their application (in terms of message exchange within a neighbourhood) especially easy. Indirect communication (of radius greater than 1 via router nodes) does not occur.

The dynamics of the mobile network of communicating units as a whole is thus a result of the concurrent activities of the individual units within the network. The activity of each individual unit is a simple process, following a simple protocol, which comprises at least the following algorithmic steps (described in pseudo-code as follows):

```
WHILE (operative) // process cycle for every unit
          {
           WHILE (repair_rule_applicable) {apply repair_rule};  // as often as possible
                 IF (move_is_desired)         {apply move_rule}; // once per cycle
                 IF (link_rule_applicable)    {apply link_rule}; // once per cycle
          };
```

In this rule-application protocol, priority is given to the application of repair-rules to maintain the desired global topological network invariants as mentioned above. Alternative designs of the protocol could obviously lead to different global network behaviour, which would make up an interesting question for experimental studies; (see Future Work section below).

4 Related Work

There are many formalisms and approaches to the modelling of distributed systems as a whole or particular aspects thereof. As far as the domain of graph transformation is concerned (which is the theme of GT-VMT-2008), a similar mobile network scenario and a similar modelling approach to it was recently published by *Casteigts* and *Chaumette* [CCh05]. It differers from the model presented in this paper as far as the embedding of geometric calculations for positions and distances into the transformation rules is concerned. *Heckel* and *Guo* described a layered graph transformation model of roaming cellphones being transferred from one base station to another one [HGu04]. Their scenario is thus similar to the scenario presented above (and also intended to be subject to simulative experiments), but geometric considerations such as network triangulation or distances between units do not play a role in their paper. Moreover, their model is developed in the classical "omnivident" paradigm, not in the paradigm of local communications. The short-paper [GHe04], published by the same authors, is little more than an abstract summary of [Hgu04] but it provides a nice description of the typical characteristics and difficulties of distributed mobile systems, as well as a nice discussion of broader related work. *Knirsch* and *Kreowski* where amongst the first ones to model agent systems on a high level of abstraction [KKr00]. Their formalism and notation appears quite similar to the one of [LMS99] but remains rather vague as far as the issue of mobility is concerned. *Chalopin*, *Godard*, *Métivier* and *Ossamy*, on the other hand, have successfully modelled agent mobility on a given network in terms of graph relabelling with message-passing semantics [CGM06], however their graph model is static and does not allow for restructurings of the network itself.

5 Future Work

To date, none of the existing graph transformation software packages offers experimental simulation environments to such an extent that hundreds of nodes of pixel-size could be displayed in motion on a computer screen. So far, existing graph transformation software packages have put all emphazise on the visualisation of structural or *topological* properties of their model graphs, not on the visualisation of their *geometric* properties and node motion, as it would be required for the communicating mobile unit scenario described in this paper. The implementation of such a software system is planned, whereby the graph transformation rules described in this paper shall be hard-coded into the prototype for the sake of runtime speed.

Once such a prototype would be available, the necessary empirical validation of the introduced concepts would be possible. Then it could be investigated through experimental observation and measurement, for example:

- How frequently does the global network invariant (triangulation) get locally violated?
- How quickly are those local violations being repaired?
- How does the size of a neighbourhood (e.g.: maximally 5 communication partners or maximally 8 per unit) as well as the length of the maximal communication-distance influence the behaviour of the entire network as a swarm-like super-unit?
- How smoothly does the network re-organize its structure when many nodes are on the move into different directions?
- How would a different rule application protocol, individually executed by each unit, affect the behaviour of the network system as a whole?
- (Etc.)

The model developed in this paper is still quite simplistic as far as the physical properties of communication links are concerned. It has been simply assumed that the maximal distance of communication is a constant D for all units in the network. In reality one would have to deal with locally variable parameters Ď, depending (for example) on weather-conditions, objects obstructing the transmission of radio signals, etc. The graph transformation model of the scenario would have to be amended accordingly – not so much in terms of the graphical rewrite-rules, but rather in terms of the functional calculations into which those rules have been embedded.

References

[BGM01] M. Bauderon, S. Gruner, Y. Métivier, M. Mosbah, A. Sellami: *Visualisation of Distributed Algorithms based on Graph Relabeling Systems*. ENTCS Vol.50, No.3, Pages 227-237, Elsevier, 2001.

[CCh05] A. Casteigts, S. Chaumette: *Dynamicity Aware Graph Relabeling Systems (DA-GRS): A Local Computation based Model to describe MANet Algorithms*. Proc. 17th IASTED Internat. Conf. on Parallel and Distr. Computing and Systems (PDCS05), pages 231-236, Pheonix, November 2005.

[CGM06] J. Chalopin, E. Godard, Y. Métivier, R. Ossamy: *Mobile Agent Algorithms vs. Message Passing Algorithms*. LNCS Vol.4305, pages 187-201, Springer, 2006.

[GHe04] P. Guo, R. Heckel: *Modeling and Simulation of Context-Aware Mobile Systems*. Proc. 19th Internat. Conf. on Automated Software Eng. (ASE'04), IEEE Computer Society, 2004.

[HGu04] R. Heckel, P. Guo: *Conceptual Modeling of Styles for Mobile Systems: a Layered Approach based on Graph Transformation*. Proc. IFIP Conf. on Mobile Information Systems (MOBIS), Oslo, September 2004.

[KKr00] P. Knirsch, H.-J. Kreowski, *A Note on Modeling Agent Systems by Graph Transformation*. LNCS Vol.1779, pages 79-86, Springer, 2000.

[LMS95] I. Litovsky, Y. Métivier, E. Sopena: *Different Local Controls for Graph Relabelling Systems*. Math. Syst. Theory Vol.28, pages 41-65, 1995.

[LMS99] I. Litovsky, Y. Métivier, E. Sopena: *Graph Relabeling Systems and Distributed Algorithms*. H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg (eds.): Handbook of Graph Grammars and Computing by Graph Transformation, Vol.3, pages 1-56, World Scientific, 1999.

[Sch96] A.. Schürr: *Logic Based Programmed Structure Rewriting Systems*. Fundamenta Informaticae Vol. 26, No. 3/4: Special Issue on Graph Transformations, pages 363-385, IOS Press, 1996.

[Sch97] A.. Schürr: *Programmed Graph Replacement Systems*. G. Rozenberg (ed.): Handbook of Graph Grammars and Computing by Graph Transformation, Vol.1, pages 479-546, World Scientific, 1997.

[SWZ99] A.. Schürr, A. Winter, A. Zündorf: *The PROGRES Approach: Language and Environment*. H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg (eds.): Handbook of Graph Grammars and Computing by Graph Transformation, Vol.2, pages 487-550, World Scientific, 1999.

Acknowledgments

During a visit to the University of Bordeaux in March 2006 I discussed several ideas related to this paper with my colleagues, particularly *Mohamed Mosbah* and *Serge Chaumette*. Thanks also to *Andy Schürr* and *Reiko Heckel* as well as three anonymous reviewers of GT-VMT-2008 for their valuable hints and suggestions.